

CSC 469H1 F
ADVANCED OPERATING SYSTEMS

UNIVERSITY OF TORONTO
Fall 2006

Term Test #2

NO AIDS ALLOWED

Please **PRINT** in answering the following requests for information:

Family Name: _____

Given Names: _____

Student Number: | _ _ _ | | _ _ _ | | _ _ _ |

Login (@cdf): _____

Notes to students:

1. This test lasts for 110 minutes and consists of 89 marks. Budget your time accordingly.
2. This test has 8 questions and 11 pages (including this one); Check that you have all pages.
3. **Write in pen. No pencils. Really, I mean it.**
4. Write your answers on this “question and answer” paper, in the spaces provided. Be concise. In general, the amount of space provided is an upper bound on the “size” of answer that is expected. If necessary, use space where available and provide explicit pointers.
5. State your assumptions and show your intermediate work, where appropriate.
6. Do **not** go beyond here until instructed to do so. Write your student number at the top of each succeeding page once you get going.

Question	Marks
1	/15
2	/10
3	/16
4	/14
5	/10
6	/12
7	/6
8	/6
Total	/89

1. [15 marks, 3 each] Definitions

Define the following terms, in the context of this course:

a) Byzantine failure

b) Clustered page table

c) Distributed Shared Memory

d) RAID Level 1

e) Vulnerability

2. [10 points] Superpages

a) **[2 points]** Any paged virtual memory system requires both hardware and software support. What changes or additions to hardware are required to support superpages?

b) **[4 points]** Describe 2 situations in which it may be desirable to *demote* a superpage.

c) **[4 points]** Explain how the use of superpages can lead to fragmentation, and what can be done about it.

3.[16 points; 4 each] Page**Placement**

Consider the following snippet of code that must run on a system with a 4kB page size and a 16kB physically-indexed direct-mapped data cache with a 16 byte cache line size.

```
...
page_size = 4096;
array_size = 2*page_size;
char *a = (char *)malloc(array_size);
char *b = (char *)malloc(array_size);

for (int i=0; i < array_size; i++) {
    a[i] = i % 256;
    b[i] = a[i] >> 1;
}
...
```

Assume the loop is the first access to the virtual memory allocated for arrays **a** and **b**, and that the variable **i** is allocated in a register (so it won't take space in the cache).

a) What is the minimum number of cache misses that will occur during the execution of this loop? What is the maximum number of cache misses?

b) Suppose that the OS is using *page coloring* for page placement, and that array **a** starts at address 0x8047000. Give an address for **b** that will minimize cache misses, and an address for **b** that will maximize cache misses. Draw a picture showing how the pages of **a** and **b** are mapped to the cache in each case.

This question continues
on the following page.

c) Assuming we don't mind wasting virtual memory space, show how you would allocate **a** and **b** (using malloc) to guarantee best case behaviour, if the OS uses page coloring. Explain the general idea in English (use a picture if you like), and then give C code. (You only need to handle this specific example - you do not have to generalize to arrays of arbitrary sizes).

d) If the OS is using *bin hopping* for page placement, will this result in the best or worst case behaviour? Draw a picture showing how the pages of **a** and **b** will be mapped to the cache in this case.

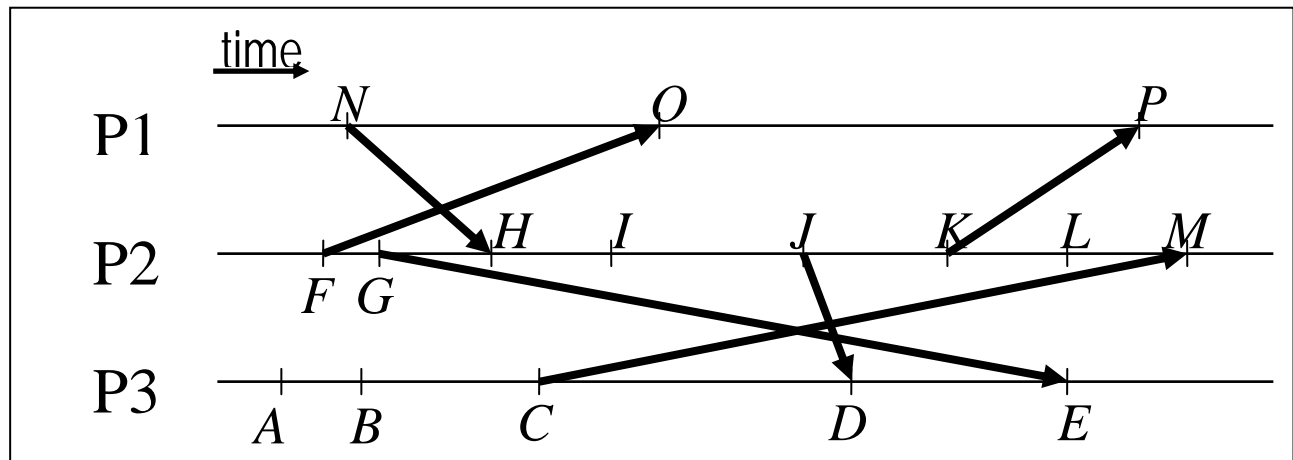
4. [14 points] Clocks in Distributed Systems

(a) [4 points] After contacting a time server, the local clock on some machine is discovered to be 4 seconds fast, with a current reading of 10:27:54.0 (hh:mm:ss).

Explain why it is undesirable to set the clock back to the correct time immediately, and show how it should be adjusted so that it is correct when it reads 10:28:0.0. Assume that the hardware timer is programmed to interrupt every 10ms.

(b) [6 points] Under Lamport's "happens-before" relation, indicate whether the following statements are true or false for the events shown on the diagram below:

- i) G happens before Q
- ii) N happens before E
- iii) D happens before P



(c) [4 points] On the diagram above, show the vector timestamps for each event.

5. [10 points] Distributed

Agreement & Group Communication

(a) [6 points] Consider the Byzantine *Generals Problem*. Show that 2 lieutenants can agree on an action (attack or retreat) based on the order given by a commanding general, even if one of the participants is faulty, *if all messages are digitally signed by the sender*.

(b) [4 points] The processes below are participating in a group communication protocol that provides *FIFO Atomic Broadcast* (meaning we combine FIFO Order and Total Order constraints). Show two possible legal orders for message delivery in P1 and P2:

P0	P1		P2		P3
	Order 1	Order 2	Order 1	Order 2	
Broadcast m1					Broadcast m3
Broadcast m2					Broadcast m4

6. [12 points; 4 each] Fault**Tolerance**

a) Explain the basic idea behind replicated state machines, and why they require atomic broadcast to provide a fault tolerant service.

b) In A3, the chatserver maintains state for each client, which is lost when the server fails and restarts. Although your clients could attempt to rebuild part of the state automatically, some problems can't be solved by the client (e.g. a new user contacting the new instance of the server and registering the user name you had with the old server before your client re-establishes a connection with the new server). Assuming you could change the server code, describe how you would improve failure recovery. Give as much detail as you can.

This question continues
on the following page.

c) Suppose you were not allowed to change the server code, but you were able to convince the maintainer of the service to run multiple instances of the chat server, and the location service was modified such that the web-hosted text file contained parameters (e.g. hostname and port numbers) for *all* currently-running instances of the chat server. (The servers themselves will not be aware that there are multiple servers running). Explain what you could do in the client code to improve fault tolerance and failure recovery.

7. [6 points] Reliable Storage

Describe how LFS handles crash recovery and ensures file system consistency.

8. [6 points] Security

Discuss why buggy programs so often lead to compromised systems, and give one example of an OS security mechanism intended to address this problem.

Extra space. Please indicate clearly which question(s) you are answering here, if any.

Total marks = (89)

End of test